Rumor detection based on tweets linguistic features

Jie Liu
jliu4@albany.edu

ABSTRACT

Due to its openness and convenience, online social medias, such as Twitter and Facebook, have become the hotbed of spreading rumors and automatically detecting rumors becomes a crucial problem, especially for twitter. Current research on rumor detection usually focuses on user features, diffusion features, and linguistic features. In this paper we talk about the drawbacks of the first two, and choose linguistic features in the classification. Two approaches are studied. In first one, we utilize a widely used software, Linguistic Inquiry and Word Count (LIWC) to find features that could distinguish rumors from non-rumors. It is followed by an SVM classifier to predict rumors. The accuracy of this approach is 87%. In second approach, the latest Convolutional Neural Networks (CNN) model is used for rumor classification and gains an accuracy of 95.5%. Finally, we compare and discuss these two methods in terms of their advantages and disadvantages.

Keywords

Rumor detection, LIWC, CNN, linguistic features.

1. INTRODUCTION

Nowadays debunking rumors has become a critical problem. On April 23 2013, for example, the official twitter handle of the respected Associated Press news agency sent out a message at about 1:07 p.m. ET, saying "Breaking: Two Explosions in the White House and Barack Obama is Injured."[3]. Although this account was quickly recovered, the tweet had spread to millions of users [4] and caused sever social panic, which led to a loss of $136.5 billion in the stock market [5]. Another example is, in [6] it is pointed out that in 2016 president election, there are 226 rumors about Hillary Clinton and 303 from Donald Trump, and fake news on social media may have influenced the election.

One method to select linguistic features is based on the frequencies of top frequently used words. There are two major disadvantages associated with this approach. First, if ready-to-use libraries are not available, it would be computationally expensive to loop over a large set of tweets texts. Second and more importantly, some useful and critical features may be lost. For example, the percentage of the ingestion words (hungry, hungrier, hungriest) in whole non-rumor tweets words set is 0.26%, while in rumor tweets it is 3.24%. This indicates that the ingestion words may be a good feature for differentiating rumor and non-rumor tweets. But because the frequency of ingestion words in whole tweets is very small, these words would not count if we only consider the most frequent words.

In contrast, utilizing LIWC2007 Dictionary to categorize words can significantly improve the efficiency when select linguistic features. The dictionary is composed of around 4,500 words, word stems and 80 output variables (descriptor categories, linguistic dimensions, personal concerns, etc.), which could be used to compute the percentage of each subcategories in the whole words. With the dictionary, 30 million words can be divided into subcategories in less than 2 minutes.

Both methods mentioned above involve selecting features manually. Selecting features is critical to the classification. But it is challenging in the sense that it is usually painstakingly detailed, biased, and labor-intensive [11]. Most of the time, it is difficult, if not impossible, for us to find all the effective features to distinct rumor and non-rumors tweets. Even though there are differences between two classes, it is not guaranteed that we are able to find a practical feature to discern them [11].

Neural network model in deep learning is an effective technique to address these problems. It uses multiple hidden layers to simultaneously choose the significant features we need, indicating that it is unnecessary to go through the time-consuming and indeterministic process of feature engineering. It is claimed that use Convolutional Neural Networks (CNN) model is efficient and effective in terms of representation [12]. In this paper we follow this idea by adopting one convolutional layer, one max pooling to do binary classification on tweets. In the experiments we achieve very good accuracy.

2. RELATED WORKS

The popular approaches to detect rumors include using user features [7,8], linguistic features [4,9] and diffusion features [1,10]. Both user features and diffusion features have its draw backs. For example, in [1] the authors observed that users participating in non-rumor spreading are more active and have larger audiences, and, on the other hand, the classification performance decreases when the observation period becomes longer. That is because rumor tweets tend to go viral, and many non-rumor accounts are also involved in spreading rumors unintentionally [1]. In contrast, in [1] employing diffusion feature achieved quite accurate results only after the rumors had been spread for a relatively long time. Linguistic features are not sensitive to the diffusion time, which is the main reason we chose it for our rumor detection task.

3. MATERIALS AND METHODS

3.1 Datasets

We find a data set posted by authors of [1]. To select rumor cases, they searched for two famous rumor archives, snopes.com and urbanlegends.about.com and identified popular rumors at the time of the dataset. For non-rumor cases, they searched for notable events from news media outlets like times.com, nytimes.com, and cnn.com. They identified a total of 130 events (72 rumors and 58 non-rumors) from the time period covered by the Twitter dataset and extracted tweets corresponding to these events based on two criteria: (1) a tweet should contain explicit keywords relevant to the event and (2) a tweet should have been posted during the time of circulation (i.e., within the first two months).

To test whether tweets identified in the above manner are indeed relevant to the event of interest, they hired four human labelers to evaluate the dataset. The labelers were asked to judge whether each event was rumor or non-rumor by examining four randomly selected tweets and URLs embedded in tweets for each event. Then, they selected the events that were evaluated by four participants and had the majority agreement for this study. The final set of rumor and non-rumor events had agreement among three or more labelers. The intra-class correlation coefficient (ICC), which measures the level of agreement, was 0.993 and the p-value was close to zero. As a result, 111 events were retained (60 rumors and 51 non-rumors).

3.2 Methods

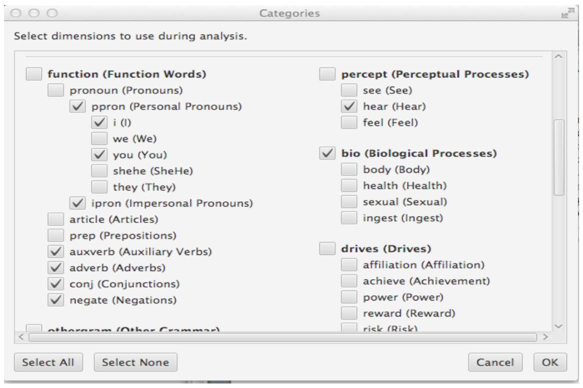
In this part, we will fully represent how we use two approaches SVM together with LIWC and CNN to select features and classify rumor and non-rumor tweets.

3.2.1 Using LIWC and SVM

"LIWC2007 contains 4 general descriptor categories (total word count, words per sentence, percentage of words captured by the dictionary, and percent of words longer than six letters), 22 standard linguistic dimensions (e.g., percentage of words in the text that are pronouns, auxiliary verbs, negative emotions, about family or friend, etc.), 32 word categories tapping psychological constructs (e.g., affect, cognition, biological processes), 7 personal concern categories (e.g., religion, leisure activities), 3 paralinguistic dimensions (assents, fillers, nonfluencies), and 12 punctuation categories (periods, commas, etc)" [13](Figure 1).

"After the processing module has read and accounted all words in a given text, it calculates the percentage of total words that match each of the dictionary categories [13]." For example, if LIWC analyze a file with 2500 words, it may find there are 100 words in perception category and 250 words in function category. Then it would be 4% for perception and 10% for function.

Figure1: perspective of subcategories of LIWC2007

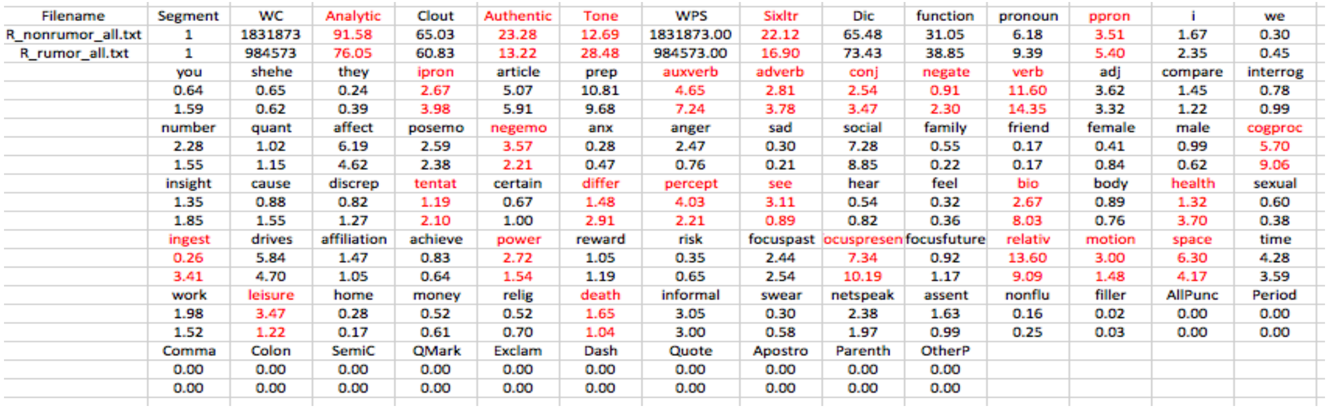


Categories

Select dimensions to use during analysis.

- [ ] function (Function Words)
  - [ ] pronoun (Pronouns)
    - [✓] ppron (Personal Pronouns)
      - [✓] i (I)
      - [ ] we (We)
      - [✓] you (You)
      - [ ] shehe (SheHe)
      - [ ] they (They)
    - [✓] ipron (Impersonal Pronouns)
  - [ ] article (Articles)
  - [ ] prep (Prepositions)
  - [✓] auxverb (Auxiliary Verbs)
  - [✓] adverb (Adverbs)
  - [✓] conj (Conjunctions)
  - [✓] negate (Negations)
- [ ] othergram (Other Grammar)
- [ ] percept (Perceptual Processes)
  - [ ] see (See)
  - [✓] hear (Hear)
  - [ ] feel (Feel)
- [✓] bio (Biological Processes)
  - [ ] body (Body)
  - [ ] health (Health)
  - [ ] sexual (Sexual)
  - [ ] ingest (Ingest)
- [ ] drives (Drives)
  - [ ] affiliation (Affiliation)
  - [ ] achieve (Achievement)
  - [ ] power (Power)
  - [ ] reward (Reward)
  - [ ] risk (Risk)

Figure2: percentage of total words that match each of the dictionary categories for files "R_nonrumor_all.txt" and "R_rumor_all.txt"



Filename	Segment	WC	Analytic	Clout	Authentic	Tone	WPS	Sixltr	Dic	function	pronoun	ppron	i	we
R_nonrumor_all.txt	1	1831873	91.58	65.03	23.28	12.69	1831873.00	22.12	65.48	31.05	6.18	3.51	1.67	0.30
R_rumor_all.txt	1	984573	76.05	60.83	13.22	28.48	984573.00	16.90	73.43	38.85	9.39	5.40	2.35	0.45
	you	shehe	they	ipron	article	prep	auxverb	adverb	conj	negate	verb	adj	compare	interrog
	0.64	0.65	0.24	2.67	5.07	10.81	4.65	2.81	2.54	0.91	11.60	3.62	1.45	0.78
	1.59	0.62	0.39	3.98	5.91	9.68	7.24	3.78	3.47	2.30	14.35	3.32	1.22	0.99
	number	quant	affect	posemo	negemo	anx	anger	sad	social	family	friend	female	male	cogproc
	2.28	1.02	6.19	2.59	3.57	0.28	2.47	0.30	7.28	0.55	0.17	0.41	0.99	5.70
	1.55	1.15	4.62	2.38	2.21	0.47	0.76	0.21	8.85	0.22	0.17	0.84	0.62	9.06
	insight	cause	discrep	tentat	certain	differ	percept	see	hear	feel	bio	body	health	sexual
	1.35	0.88	0.82	1.19	0.67	1.48	4.03	3.11	0.54	0.32	2.67	0.89	1.32	0.60
	1.85	1.55	1.27	2.10	1.00	2.91	2.21	0.89	0.82	0.36	8.03	0.76	3.70	0.38
	ingest	drives	affiliation	achieve	power	reward	risk	focuspast	ocuspresen	focusfuture	relativ	motion	space	time
	0.26	5.84	1.47	0.83	2.72	1.05	0.35	2.44	7.34	0.92	13.60	3.00	6.30	4.28
	3.41	4.70	1.05	0.64	1.54	1.19	0.65	2.54	10.19	1.17	9.09	1.48	4.17	3.59
	work	leisure	home	money	relig	death	informal	swear	netspeak	assent	nonflu	filler	AllPunc	Period
	1.98	3.47	0.28	0.52	0.52	1.65	3.05	0.30	2.38	1.63	0.16	0.02	0.00	0.00
	1.52	1.22	0.17	0.61	0.70	1.04	3.00	0.58	1.97	0.99	0.25	0.03	0.00	0.00
	Comma	Colon	SemiC	QMark	Exclam	Dash	Quote	Apostro	Parenth	OtherP				
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				

After getting rid of all the special characters, emojs and links from the tweet text, we put all non-rumor tweets in the file "R_nonrumor.txt" and rumor tweets in file "R_rumor.txt". Then we load them into LIWC2007, which computes and outputs the percentages of words in files "R_nonrumor_all.txt" and "R_rumor_all.txt" that match each category (Figure 2). After that, we choose features that could efficiently differentiate rumor and non-rumor tweets with such a standard: the larger percentage for one feature must be at least 30% greater than the smaller percentage; the difference of these two percentages must be greater than 1%. Here 30% serves as a measurement to discern rumor and non-rumor tweets while 1% guarantees that the feature words are frequently occurred words. Red colored features in figure2 are the ones we choose according to our standards.

After selecting features, we set categories of LIWC2007 only to the features we choose. We put tweets of the same topic in one file and load all topic files into LIWC2007, which generates 111 rows × 28 columns excel file named "Linguistic_trainData.xlsx" as our train data.

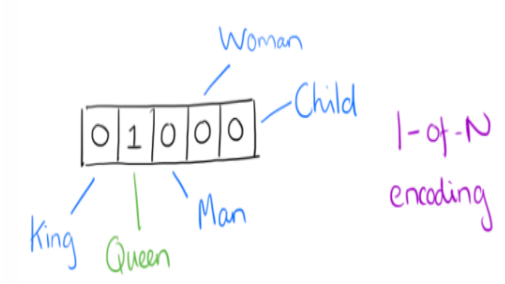
Finally, SVM and ten-folder cross validation are used to analyze this data. It achieves a very good accuracy 87%, which means features we choose for linguistic are very efficient features.

3.2.2 Using CNN

Two basic knowledge needs to be known to understand Convolutional Neural Networks. First is word to vector embedding.

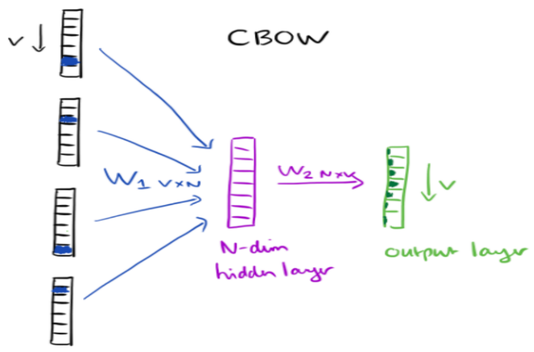
Figure 3 is the demonstration of one way to conversing one word to an embedded matrix. In paper [15] they use the encoding method of one-hot vector and each vector is associated one specific word in the dictionary. One-hot vector is a V dimension

Figure 3: one hot vector representation of word Queen [14]



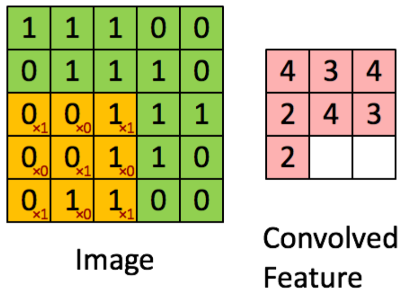
vector with one element is 1, others all 0. By applying this method, we can get the vector representation of each word. In paper [14], the authors bring out continuous bag-of-word model (CBOW) (Figure 4) and the following understanding of this model, which can translate one tweet into one matrix. Input layer is a single word that is represent by the one-hot vector x_v. By multiplying with weight matrix W, input vector x_v change to a 1× N matrix h_N in Hidden layer. Then they applying h_N a multiplication with another weight matrix W' which is N×V dimension, finally change h_N into a V× 1 vector y_v in output layer. Using this method, we change each tweet in our file into a words matrix.

Figure 4: A simple CBOW model with only one word in the context [14]



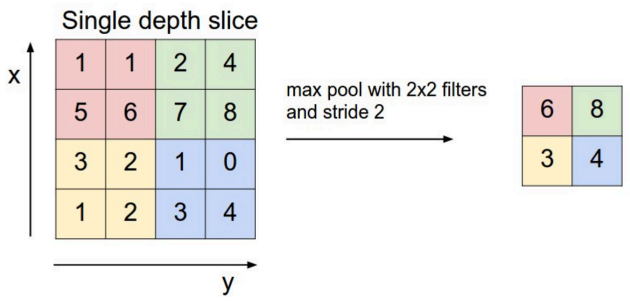
Second part is convolutional neural network part. First concept is convolution. In Fig 5, the yellow colored part combining with small red numbers is a filter. Multiplying the red numbers with the black colored elements in the matrix and then summing them up, we get one of our convolved features, which is 2 for this step in this figure. Applying this filter on all the 3 × 3 area in this matrix, we get the convolved feature matrix.

Figure 5: convolution [12]



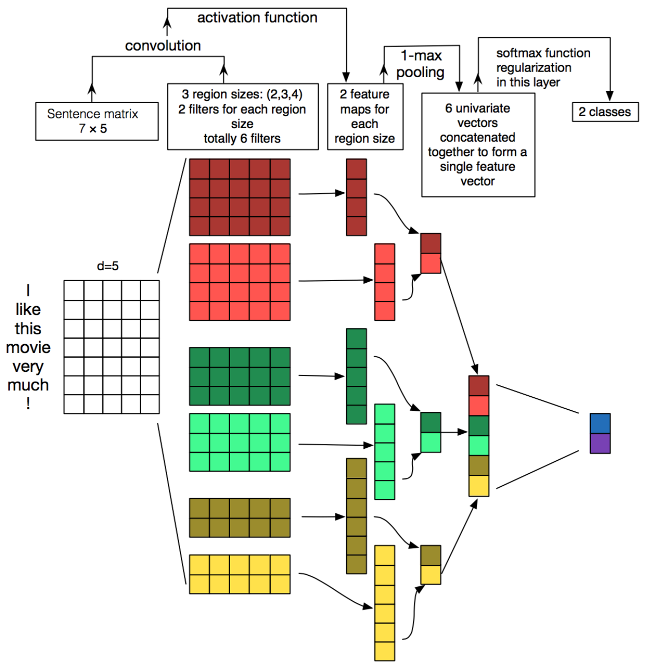
Second concept is pooling. Max pooling is the most common way to do pooling, which apply a max operation to the result of filter process [2]. For example, in figure 6, they apply a 2 × 2 filter on the matrix, picking the maximum number on each 2 × 2 area in the matrix and finally resulting in a 2 × 2 output matrix.

Figure 6: max-pooling [12]



Now, we can talk about convolutional neural network architecture in figure 7. I learn CNN architecture information (this whole paragraph) from this webpage [2] [12]. In this figure, the input is a sentence matrix and they apply 3 region sizes (2,3,4), 2 filters for each region, which are total 6 filters on the sentence matrix. Then "every filter performs convolution on the sentence matrix to get a feature map". After that they use 1-max pooling on each feature map to save the most significant features. In the end, they concentrate these 6 features into one 6-dimension vector. The final softmax layer then receives this feature vector as input and uses it to classify the sentence; here they assume binary classification and hence depict two possible output states.

Figure 7: Illustration of Convolutional Neural Network (CNN) architecture for sentence classification [12]


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

These are the whole concepts and process to use convolutional neural network to do rumor detection. After that I look through many code in Gitbub and find the code posted by user called dennybritz matches this problem best. Then I modify this code to fit in this project data. After it has been run for 8 hours, it achieves an accuracy of 95.5%.

4. EXPERIMENT RESULTS

In this paper, we use tweets of same topic as an input file. The accuracy of using features selected by LIWC2007 is 87%, which is a very good score. But if we use single tweet as an input file, the accuracy decreased to 73%. Maybe because the number of total words in one single tweet is small, the resulted data matrix is too sparse.

For CNN model, it is really convenient since we don't need to select the features manually and it gets a very good accuracy 95.5%. But building a neural network is not an easy task, which is totally different according to different conditions.

5. DISCUSSION

Neural network is a very hot topic in machine learning area these days. There are a lot of medical data, most of which are not well observed. At the same time, mining the medical data means a lot to human health and happiness. I have gained more knowledge in this area after this paper. Therefore, next step I plan to use convolutional neural network or recurrent neural network to mining and analyzing massive medical data.

6. REFERENCES

[1] Kwon S, Cha M, Jung K (2017) Rumor Detection over Varying Time Windows. PLoS ONE 12(1): e0168344. https://doi.org/10.1371/journal.pone.0168344

[2] Source: http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/ retrieved on 04/16/2017

[3] Source:http://www.telegraph.co.uk/finance/markets/10013768/Bogus-AP-tweet-about-explosion-at-the-White-House-wipes-billions-off-US-markets.html retrieved on 04/03/2017

[4] Mei, Q., Resnick, P., & Zhao, Z. (2015). Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. *WWW*.

[5] Chen, Tong and Wu, Lin and Li, Xue and Zhang, Jun and Yin, Hongzhi and Wang, Yang (2017) Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. arXiv preprint arXiv:1704.05973

[6] Cao, J., Guo, H., Jin, Z., Luo, J., Wang, Y., & Zhang, Y. (2017). Rumor Detection on Twitter Pertaining to the 2016 U.S. Presidential Election. *CoRR, abs/1701.06250.*

[7] Z. Chu, S. Gianvecchio, H. Wang and S. Jajodia, "Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?," in *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 811-824, Nov.-Dec. 2012.

[8] Cao, X.,(2015) Detecting Clusters of Fake Accounts in Online Social Networks. Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. Pages 91-101

[9] Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig . Linguistic regularities in continuous space word representations In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (June 2013), pp. 746-751.

[10] Zhiwei J., Juan C., Yu-Gang J., Yongdong Z., "News Credibility Evaluation on Microblog with a Hierarchical Propagation Model", vol. 00, no, pp. 230-239, 2014, doi:10.1109/ICDM.2014.91

[11] Cha, M., Gao, W., Jansen, B.J., Kwon, S., Ma, J., Mitra, P., & Wong, K. (2016). Detecting Rumors from Microblogs with Recurrent Neural Networks. *IJCAI*.

[12] Source: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/ retrieved on 04/16/2017

[13] Booth, R.J., Chung, C.K., Gonzales, A., Ireland, M., & Pennebaker, J.W. (2007). The Development and Psychometric Properties of LIWC2007.

[14] Rong, X. (2014). word2vec Parameter Learning Explained. *CoRR, abs/1411.2738*.

[15] Source: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/ retrieved on 04/15/2017

[16] Source: https://github.com/dennybritz/cnn-text-classification-tf retrieved on 04/16/2017